ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Laboratorio di

Sicurezza Informatica

Esercitazione: SSH tunnels e Rete Tor

Andrea Melis

Marco Prandini

Dipartimento di Informatica – Scienza e Ingegneria

Agenda

- Socks5 vs Http Proxy
  - Differenze
- Ssh tunnel
  - Perchè servono
  - Tipi di tunnel
- Tor
  - Cosa è
  - Come funziona
  - Proxy via tor
  - Tor Browser
  - Mini intro sul “dark-web”

Socks5 vs Proxy (http)

- A differenza dei proxy HTTP, che possono interpretare e funzionare solo con pagine Web HTTP e HTTPS, i proxy SOCKS5 possono funzionare con qualsiasi tipo traffico.

- I proxy HTTP sono proxy di alto livello solitamente progettati per un protocollo specifico. Sebbene ciò significhi ottenere velocità di connessione migliori, non sono così flessibili e sicure come i proxy SOCKS. I proxy SOCKS sono proxy di basso livello in grado di gestire qualsiasi programma o protocollo e qualsiasi traffico senza limitazioni.

- Su un proxy HTTP è molto più semplice poter vedere i dati delle richieste, anche utilizzando un canale TLS (che è sicuramente un molto più sicuro che in chiaro) è possibile fare attacchi di MiTM.

- Socks5 può essere invece associato ad una connessione ssh

Socks5 vs Proxy (http)

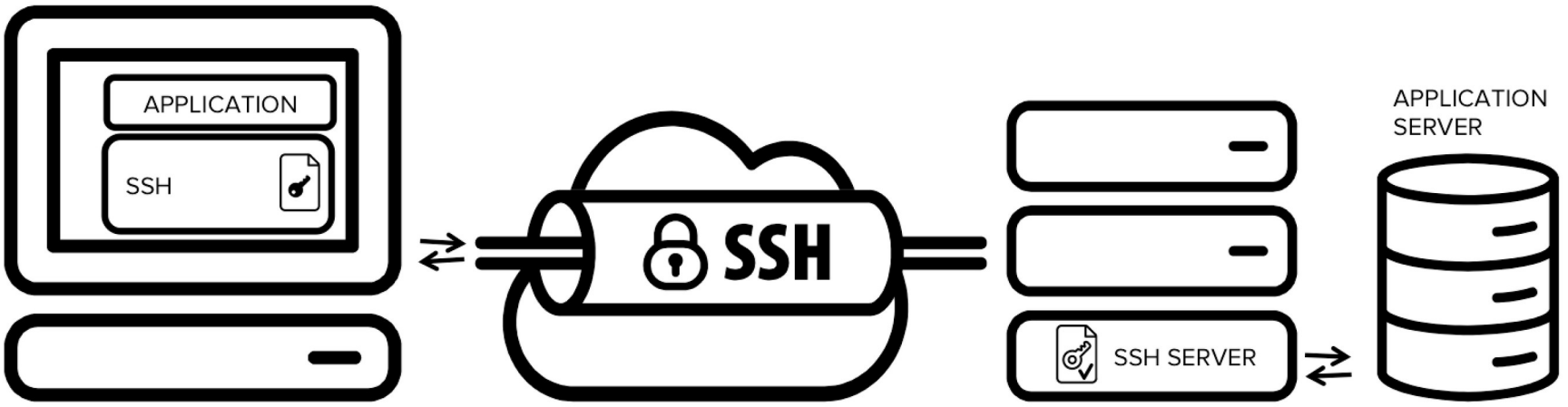
- Ricapitoliamo le differenze principali:

	Sock5	HTTP Proxy
Supporto	Teoricamente qualsiasi protocollo TCP/UDP	HTTP/(S)
Sicurezza	Supporta autenticazione, associabile a ssh	Supporta autenticazione, Soggetto a MiTM
Performance	Genericamente elevate	Elevate, tranne i proxy pubblici
Configurazione	Semplice (ssh)	Può richiedere dei tool aggiuntivi

Che differenza c'è invece con socks4? È come socks5 ma non supporta l'autenticazione

Tunnel ssh

- Il tunneling SSH è un metodo per comunicare dati su una connessione SSH crittografata.
- È principalmente utilizzato per aggiungere uno strato di crittografia alla comunicazione di un'applicazione.
- Oppure per "scavalcare" un firewall che permette la comunicazione soltanto attravero determinate porte.



Tunnel ssh

- SSH è uno standard per accessi remoti sicuri e trasferimenti di file su reti non affidabili. Fornisce inoltre un modo per proteggere il traffico dati di una determinata applicazione utilizzando il port forwarding
- Il port forwarding consiste nel tunneling di qualsiasi porta TCP / IP su SSH. Ciò significa che il traffico dati dell'applicazione viene reindirizzato all'interno di una connessione SSH crittografata in modo che non possa essere intercettato o intercettato mentre è in transito.
- Il tunneling SSH consente di aggiungere la sicurezza di rete alle applicazioni legacy che non supportano in modo nativo la crittografia.

Svantaggi

- Gli svantaggi consistono nel fatto che l'utente ha bisogno di potersi autenticare sulla destinazione. Un attaccante può quindi sfruttare la stessa "feature" per effettuare un accesso malevolo.
- È quindi necessario configurare in maniera sicura il vostro server ssh.
- Cosi come un utente qualsiasi lo stesso attaccante può creare un tunnel ssh che funga da "backdoor".

Vantaggi

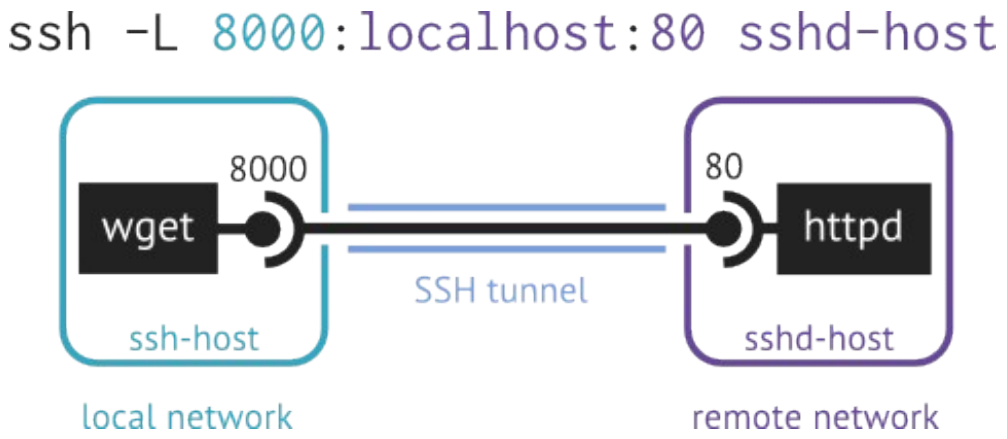
- I tunnel SSH sono ampiamente utilizzati in molti ambienti aziendali.
- In questi ambienti le applicazioni stesse possono avere un supporto nativo molto limitato per la sicurezza. Utilizzando il tunneling, è possibile ottenere la conformità con SOX, HIPAA, PCI-DSS e altri standard senza dover modificare le applicazioni.
- Questo perché modificare le applicazioni può non essere una azione semplice da compiere.
  - Necessità di riconfigurare altri workflow
  - Mancanza del codice sorgente
  - Problemi legacy

SSH Tunnel

- Esistono fondamentalmente 3 tipi di tunnel
  - Local port forwarding
  - Remote port forwarding
  - Dynamic forwarding

Local Port Forwarding

- Il Local Port Forwarding viene utilizzato per inoltrare una porta dalla macchina client alla macchina server.
- Fondamentalmente, il client SSH ascolta le connessioni su una porta e quando riceve una connessione, esegue il tunneling della connessione a un server SSH.
- Il server si connette a una porta di destinazione configurata, possibilmente su una macchina diversa dal server SSH.



Local Port Forwarding

- Gli usi tipici per il port forwarding locale includono:
  - Tunneling di sessioni e trasferimenti di file
  - Connessione a un servizio su una rete interna dall'esterno
  - Connessione a un shared-file remota
- Facciamo un esempio. Per prima cosa istanziamo un web server sulla macchina del laboratorio visibile soltanto dall'interno della macchina stessa.
- Dalla home di sec lanciamo:

```
mkdir local_server
cd local_server
cp /var/www/html/index.nginx-debian.html index.html
nano index.html
aggiungiamo al titolo dell'index.html local nginx!
```

Local Port Forwarding

- A questo punto lanciamo lanciamo:

```
python3 -m http.server --bind 127.0.0.1 8080
Serving HTTP on 127.0.0.1 port 8080
(http://127.0.0.1:8080/) …
```

- In questo modo abbiamo creato un piccolo web server in ascolto sulla porta 8080 ma SOLTANTO da localhost, che significa?

- Signifca che dalla vostra macchina principale non potete accedere al sito. Navigando infatti dalla vostra macchina Guest all'indirizzo:
  http://$ip_rete_host_only_macchina_lab:8080
  non vi comparirà nessun indirizzo

Local Port Forwarding

- A questo punto per possiamo creare un tunnel locale ssh
- L'idea è di impostare il forward della porta locale 8080 alla porta 8080 del server.
- Dalla vostra macchina guest eseguite:

```
ssh -L 8080:localhost:8080 sec@192.168.56.XX
```

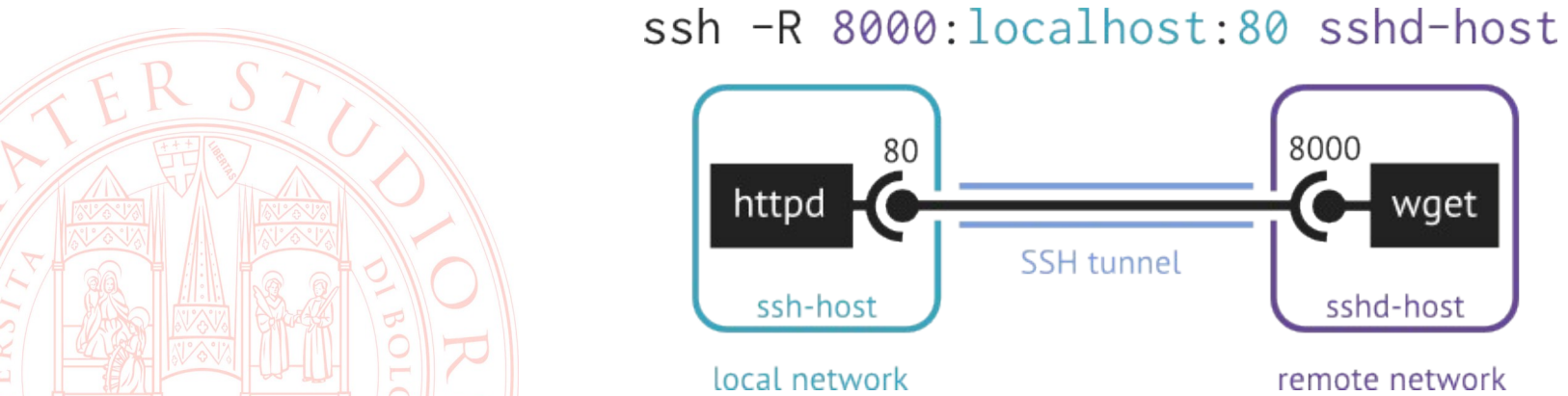
- -L è la direttiva per impostare il local port forwarding, in questo modo definiamo il forward della porta locale 8080 alla porta 8080 del server.

- A questo punto navigando col vostro browser sulla macchina guest all'indirizzo:
  http://localhost:8080

  vedrete la pagina principale del sito!

Remote Port Forwarding

- Remote port forwarding è l'opposto dell'inoltro locale e non viene utilizzato con la stessa frequenza.
- Consente di rendere disponibile una risorsa dal proprio client locale sul server SSH.
- Ad esempio, supponiamo il caso opposto a quello precedentemente affrontato.
- Supponiamo che abbiate in esecuzione localmente un server web. Quello che possiamo fare è creare un tunnel ssh in modo tale che una porta remota del server venga redirezionata ad una porta locale, permettendoci quindi di esporgli il nostro server web



Remote Port Forwarding

- Immaginando quindi di avere un web server esposto localmente sulla porta 8080 eseguiamo dalla vostra macchina guest eseguite:

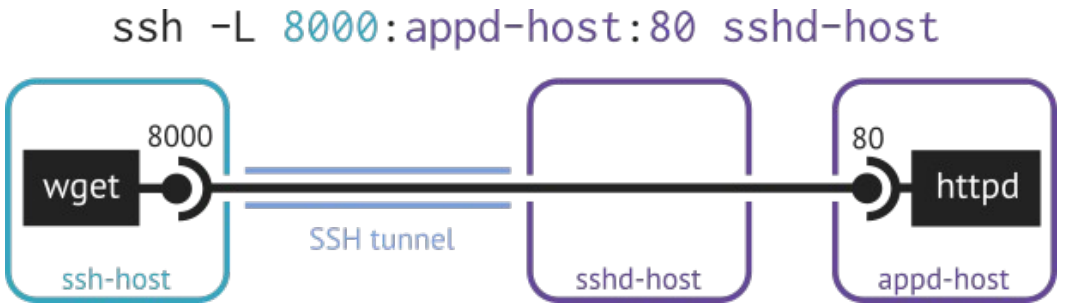
```
ssh -R 8080:localhost:8080 sec@192.168.56.XX
```

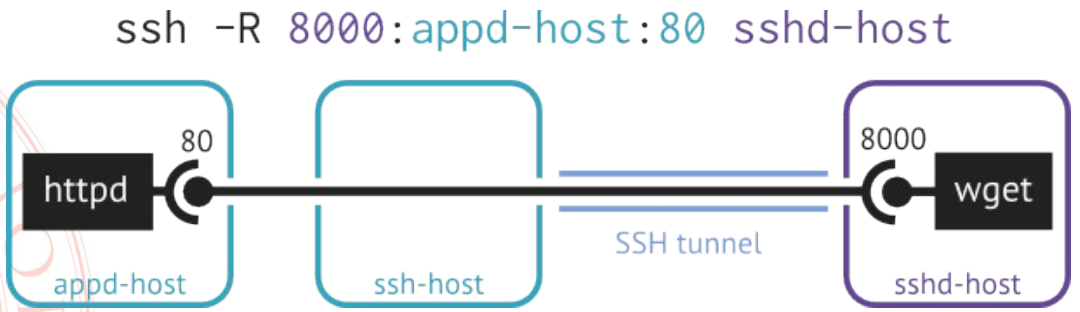
- -R è la direttiva per impostare il remote port forwarding, in questo modo definiamo il forward della porta remota 8080 alla porta 8080 del nostro client.

- Esercitazione. Creare una risorsa locale in ascolto su una porta locale (e.g webserver, serverftp, ecc) e renderla accessibile attraverso un tunner RPF alla macchina del laboratorio!

Catena di host.

- Per entrambe le tipologie di forwarding è possibile aggiungere un altro host, e fare passare il tunnel in una catena di host.
- Oppure anche solo specificarne un altro invece del localhost





Tunnel Dinamico

- Il port forwarding dinamico consente di creare un socket sulla macchina locale (client ssh), che funge da server proxy SOCKS. Quando un client si connette a questa porta, la connessione viene inoltrata alla macchina remota (server ssh), che viene quindi inoltrata a una porta dinamica sulla macchina di destinazione.

- In questo modo, tutte le applicazioni che utilizzano il proxy SOCKS si connetteranno al server SSH e il server inoltrerà tutto il traffico alla sua destinazione effettiva.

Tunnel Dinamico

- In Linux, macOS e altri sistemi Unix per creare un port forwarding dinamico (SOCKS) passare l'opzione -D al client ssh:

```
ssh -D 8080 sec@192.168.56.XX
```

- Impostare poi il browser(?) per usare il tunnel come proxy.

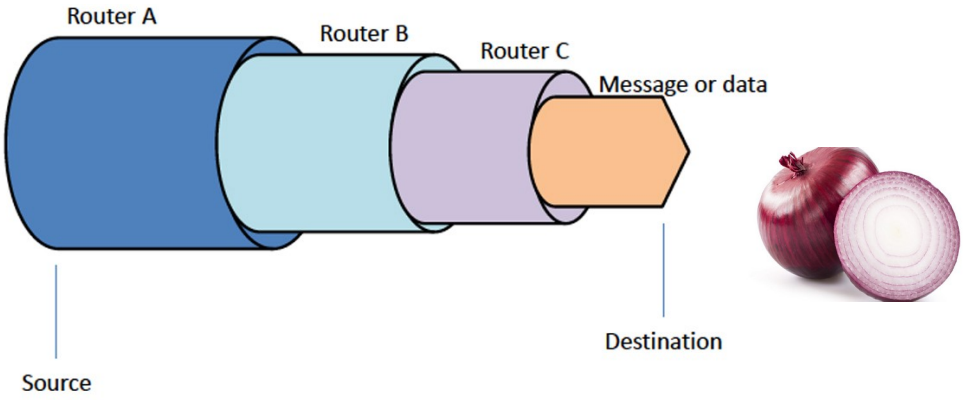
- Esercitazione per casa. Provate a creare un tunnel ssh su una macchina con ip fuori dalla vostra rete. Esempio:
  - Istanza Aws free tier
  - https://opentunnel.net/

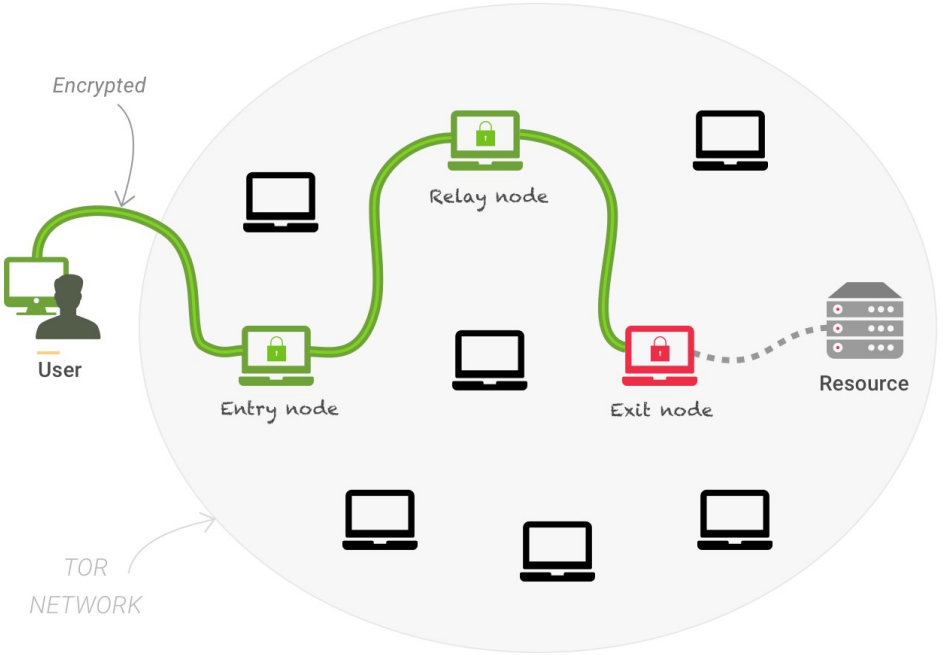
Tor project

- Progetto nato alla fine degli anni 90 per anonimizzare le comunicazioni dei servizi segreti statunitensi
- Diventato presto progetto open-source fortemente collaborativo
- Basato sul concetto di protocollo di rete “a cipolla”: Onion Routing
- Non fornisce anonimizzazione sicura al 100%! Ma solo più complicato risalire all’identità (non esiste anonimizzazione al 100%)
- Ancora molto popolare al giorno d’oggi per traffici leciti, ma soprattutto illeciti.

Onion Routing

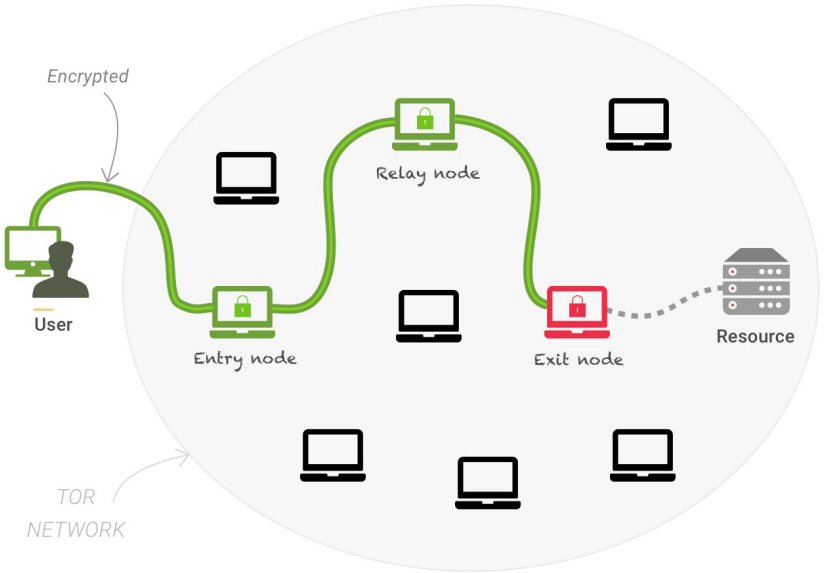
- Quando ci si connette alla rete tor si ottiene una serie di nodi della rete che attraverserete, da un nodo di ingresso, nodi intermedi e un nodo di uscita
- Ogni nodo possiede una chiave specifica, quello che fate è quindi crittare la comunicazione (pacchetto) con tutte le chiavi dei nodi della rete tor assegnati, in ordine di instradamento
- Ogni nodo riceve il pacchetto, sa decifrarlo perché rappresenta la cifrazione più “esterna” e vede cosi a quale nodo successivo inviarlo, non può però vedere il dato in chiaro.
- Qualsiasi nodo intermedio vede solo da dove è arrivato il pacchetto e dove deve instradarlo. Non sa se è un nodo intermedio e non può leggere il contenuto del pacchetto





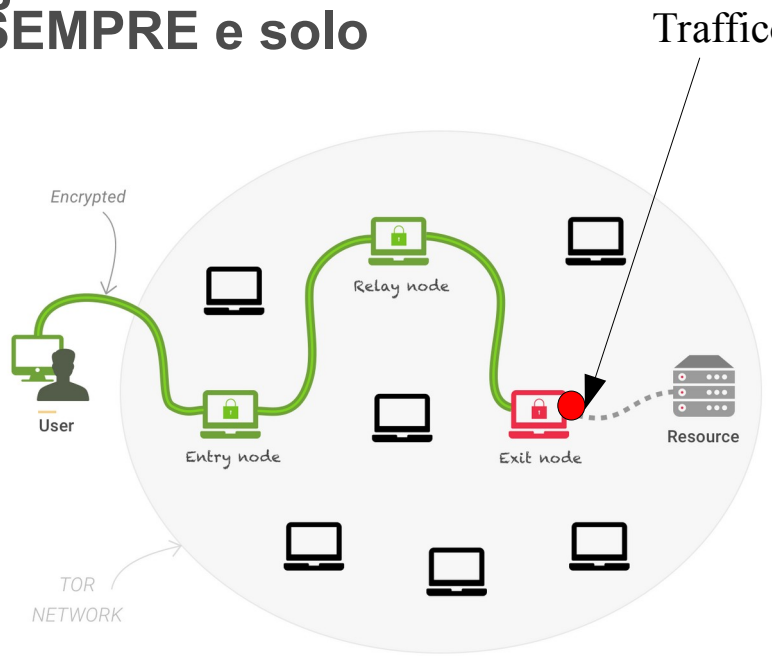
Onion Routing

- Attualmente ci sono all'incirca ~7000 nodi nella rete Tor. Teoricamente qualsiasi persona/device può far parte della rete, offrendo proprio banda e diventando un nodo.
- Potete diventare nodi intermedi, o nodi di uscita, in entrambi i casi i nodi sono comunque pubblicamente noti.
- Esistono anche dei nodi particolari chiamati Bridge Node fatti per aggirare politiche di censura di stati dove il controllo del traffico è molto invasivo (e.g. Cina) per maggiori info https://bridges.torproject.org/
- Anche se pensate di essere anonimi utilizzando Tor è bene rispettare una netiquette e fare attenzione ad alcuni problemi



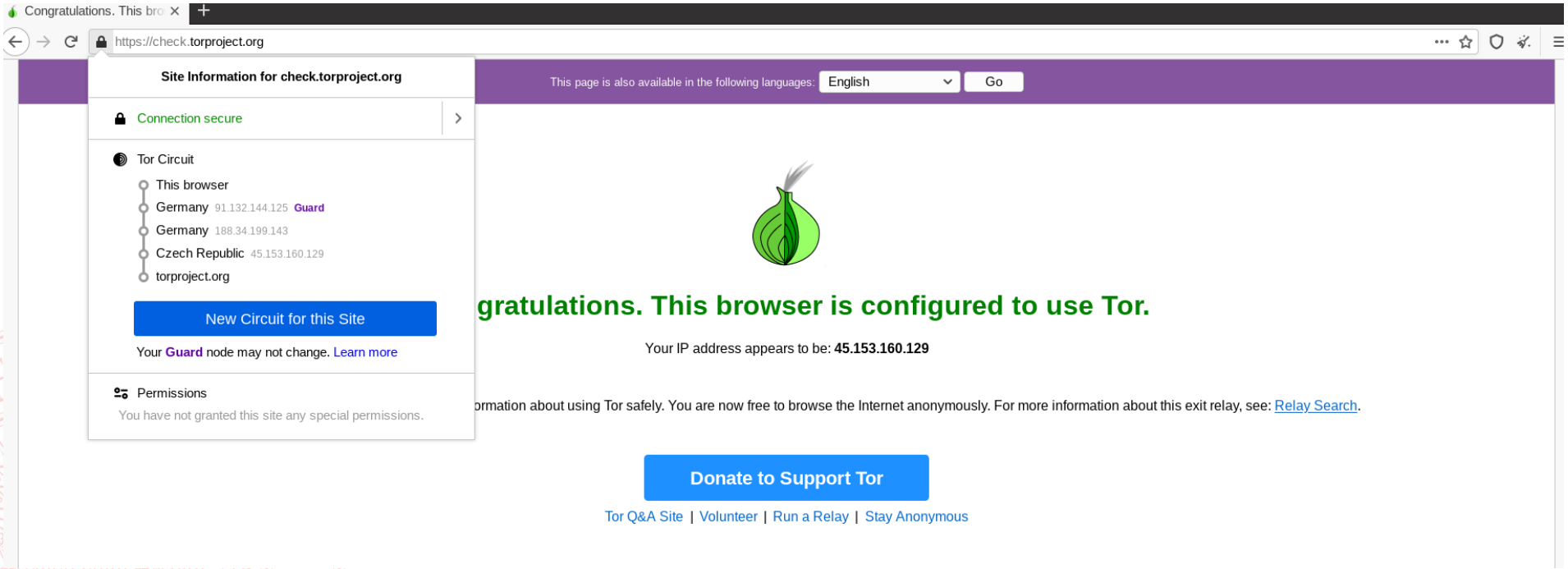
Problemi

- Traffico Diretto. Particolare traffico, o applicazione se non configurata in modo opportuno può NON passare attraverso Tor, ed essere quindi itilizzata per rompere il vostro anonimato, esempio:
  - DNS. Le richieste DNS se non specificato dal proxy Tor viaggiano in "chiaro"
  - Flash. Lo stream attraverso flash player non passava dal tor proxy, evadendo l'anonimato. Per fortuna anche grazie alla sua estrema vulnerabilità flash è stato FINALMENTE abbandonato
  - Nodo di uscita. Il nodo di uscita essendo l'ultimo nodo della catena dei nodi Tor che avete attraversato è in grado di vedere il traffico dell'utente! È bene quindi utilizzare SEMPRE e solo connessioni crittate come TLS. In aggiunta a questo anche con TLS è possibile che il nodo di uscita possa effettuare un attacco MITM e decifrare il traffico utente per cui è bene comunque fare attenzione



Utilizzo

- Si può utilizzare Tor in due modi
- 1) Se volete utilizzare Tor per la navigazione internet è sufficente scaricare il browser già completamente predisposto che offre lo stesso Tor-Project chiamato Tor-Browser (disponibile anche per mobile) https://www.torproject.org/download/
- 2) Una seconda opzione è quella di connettersi a Tor, e poi usare un Proxy che redirezioni il traffico di un applicativo sulla rete Tor.



Specificando il proxy

- Sulla macchina virtuale del laboratorio, la rete tor è già installa e attiva. Non c'è bisogno di fare nulla ma si può verificare lo stato ed eventualmente riavviarlo con:

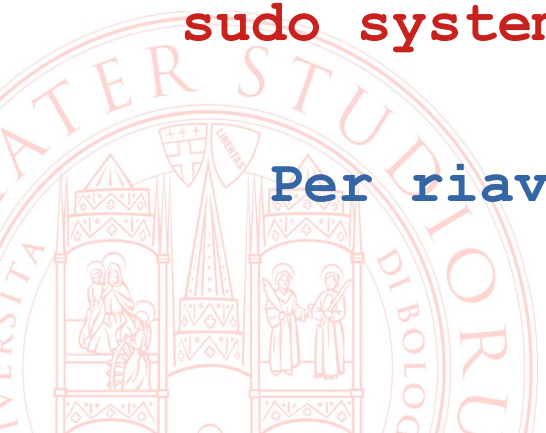
```
sudo systemctl status tor.service
```

```
tor.service - Anonymizing overlay network for TCP (multi-instance-master)
   Loaded: loaded (/lib/systemd/system/tor.service; enabled; vendor preset: enabled)
   Active: active (exited) since Fri 2021-04-09 18:10:33 GMT; 12min ago
  Process: 1823 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
 Main PID: 1823 (code=exited, status=0/SUCCESS)

Apr 09 18:10:32 sec systemd[1]: Starting Anonymizing overlay network for TCP (multi-instance-master)...
Apr 09 18:10:33 sec systemd[1]: Started Anonymizing overlay network for TCP (multi-instance-master).
```

```
sudo systemctl restart tor.service
```

Per riavviare il servizio ed avere nuovi nodi



Specificando il proxy

- Abbiamo quindi la connessione attiva verso la rete tor, è necessario però avere un modo per redirezionare il traffico, a tor, e in generale ad un qualsiasi proxy che avete impostato o ottenuto.
- Per fare questo utilizziamo proxychains. Il funzionamento è abbastanza semplice, è sufficente impostare il redirezionamento dal file di configurazione.

```
sudo nano /etc/proxychains.conf
```

- Il file di configurazione è già impostato di default per lavorare con tor, le opzioni però importanti da abilitare sono:

```
random_chain per specificare di selezionare ogni volta i nodi
in maniera casuale (abilitate questo e disabilitate strict_chain
che fa l'opposto)

...

#proxy_dns per specificare che anche le richieste dns debbano
passare dal proxy

...

socks5 127.0.0.1 9050 (user) (pass) per specificare il
proxy come tipo, indirizzo, porta, eventuali_cred (la porta 9050
in localhost è quella dove "gira" tor
```

Specificando il proxy: Web Test

- Salvate il file e facciamo il primo test.
- Verifichiamo di essere in grado di navigare su internet attraverso la rete tor.
- Utilizziamo quindi firefox istruendolo affinché passi da tor utilizzando proychains, lanciamo quindi.

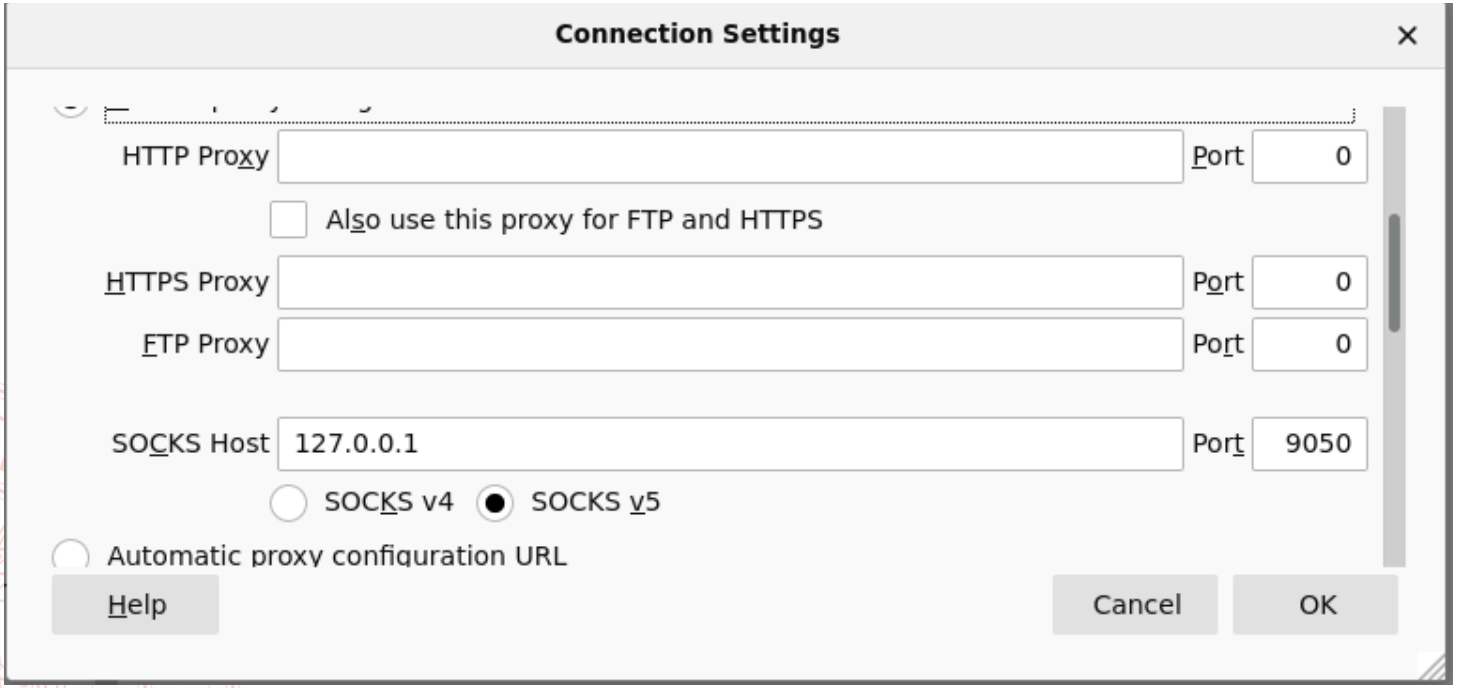
```
proxychains firefox "https://check.torprojects.org"
```

- Il sito verifica che si stia navigando dalla rete tor e dovrebbe avere successo!
- È possibile fare la controprova lanciando firefox senza proxychains allo stesso URL.
- Proxychains col comando precedente mostra anche tutti gli "hop" che compie per arrivare a destinazione

```
ProxyChains-3.1 (http://proxychains.sf.net)
|DNS-request| detectportal.firefox.com
|DNS-request| check.torproject.org
|R-chain|-<>-127.0.0.1:9050-<><>-4.2.2.2:53-|R-chain|-<>-127.0.0.1:9050-
<><>4.2.2.2:53-|DNS-request| content-signature-2.cdn.mozilla.net
|R-chain|-<>-127.0.0.1:9050-<><>-4.2.2.2:53-<><>-OK
<><>-OK
<><>-OK
… NOTARE SIA LE RICHIESTE TRA I NODI CHE QUELLE DNS!
```

Specificando il proxy: Web Test .. easier

- Con i browser moderni non c'è però bisogno di utilizzare tool aggiuntivi per reindirizzare il traffico su un proxy, ma possiedo già tutte le opzioni disponibili
- Senza utilizzare proxychain è infatti possibile ottenere lo stesso risultato lanciando firefox → Preferences → Network Settings→ Settings e impostiamo manualmente il sock5 proxy su 127.0.0.1 porta 9050.
- Navigando nuovamente su `https://check.torprojects.org` saremo nuovamente connessi a tor



Usare Tor NON per Web Browsing

- Seguendo lo stesso meccanismo utilizzato con firefox è possibile usare proxychains per redigere il traffico di qualsiasi altro tool.
- Prendiamo ad esempio sqlmap. Sqlmap è un tool per automatizzare attacchi di Sql Injection che vedremo più avanti.
- L'unica cosa che dovete sapere è che effettua delle richieste Http ad un'applicazione vulnerabile.
- Possiamo usare tor per lanciare l'attacco in forma anonima, oppure per evitare di essere messi in blacklist.
- Lanciamo un piccolo test con sqlmap col comando:

```
proxychains sqlmap -u "http://www.target.com/abc.php?cat=50"
```

```
|DNS-request| www.target.com
|R-chain|-<>-127.0.0.1:9050-<><>-4.2.2.2:53-<><>-OK
|DNS-response| www.target.com is 151.101.14.187
[21:52:12] [INFO] testing connection to the target URL
|DNS-request| www.target.com
|R-chain|-<>-127.0.0.1:9050-<><>-4.2.2.2:53-<><>-OK
|DNS-response| www.target.com is 151.101.14.187
```

Il comando testa il parametro vulnerabile "cat" sul sito target.com, e come per firefox notiamo anche qui come le richieste, DNS incluse, passano attraverso tor.

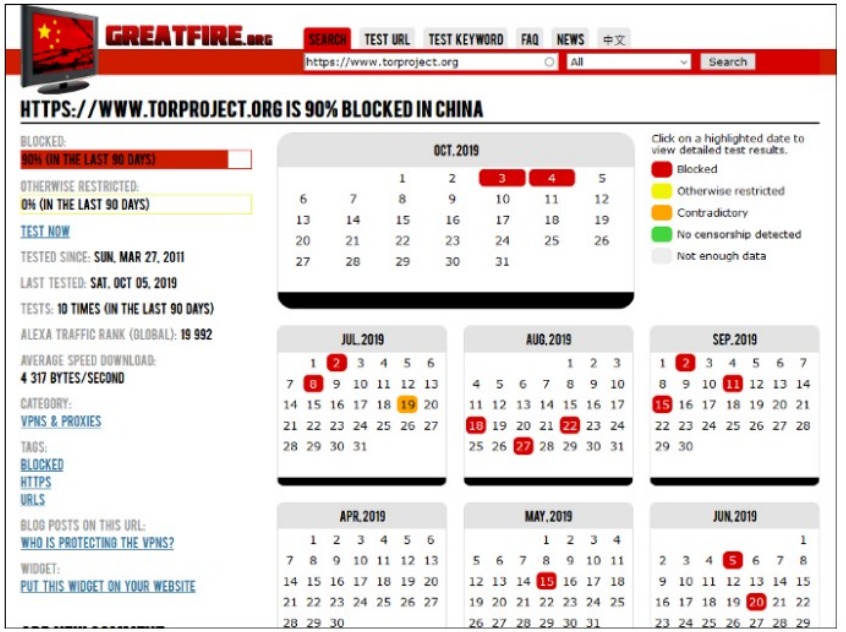
Tor nella vita reale: Censura e Dark Web

- Al giorno d'oggi Tor è utilizzato per due principali scopi:
  - Combattere la censura
  - Il deep web

- In molti paesi (e.g. Cina) tor è utilizzato come proxy o per anonimizzare il traffico per evadere i limiti e il controllo delle attività governativa.

- Essendo noti i nodi e la rete viene molto spesso bloccata, ma attraverso i bridge node o altri bridge dedicati le statistiche indicano che almeno 2000 nodi siano comunque sempre attivi anche dalla Cina

  fonte: https://arxiv.org/pdf/1204.0447v1.pdf

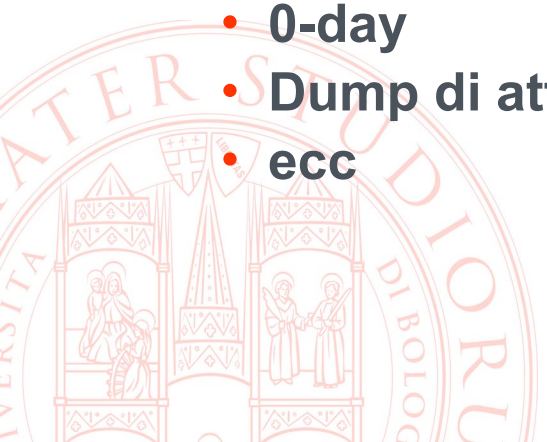


Tor e Dark Web

- L'utilizzo principale di tor tuttavia rimane quello di essere la chiave di accesso al cosidetto "Dark Web"
- Per Dark Web si intendono una parte di contenuti non indicizzati (facenti quindi parte del Deep Web) che è possibile raggiungere soltanto attraverso reti come quella tor.
- Essendo, come detto, i nodi della rete tor pubblici è possibile creare dei web server non indicizzati e accedibili soltanto se vi si arriva da un nodo tor.
- La più tipica rappresentazione è la seguente:

Tor e Dark Web

- Nel mondo del dark web c'è ovviamente di tutto
  - Attività non necessariamente criminali
    - Attivismo politico
    - Attivismo sociale
    - Scambio dati e info protetti
  - Attività decisamente criminali
    - Compravendita droga
    - Compravendita armi
    - Terrorismo
    - Pedopornografia
    - Informazioni personali
    - 0-day
    - Dump di attacchi
    - ecc



Tor e Dark Web

- Il mondo dei mercati sul dark web è sicuramente un argomento molto interessante.
- I mercato sul dark web sono sempre esistiti, ma fino a qualche anno fa le compravendite erano complicate ed erano necessari alcuni stratagemmi per poter effettuare i pagamenti in modo anonimo ad esempio:
  - Prepagate one-shot anonime
  - Money Transfer
  - Ecc

- L'era della crypto currencies ha sicuramente dato un impulso enorme a questi mercati che si stima al giorno d'oggi abbiano raggiunto un volume di più di 2 miliardi di dollari [fonte: https://www.nature.com/articles/s41598-020-74416-y]

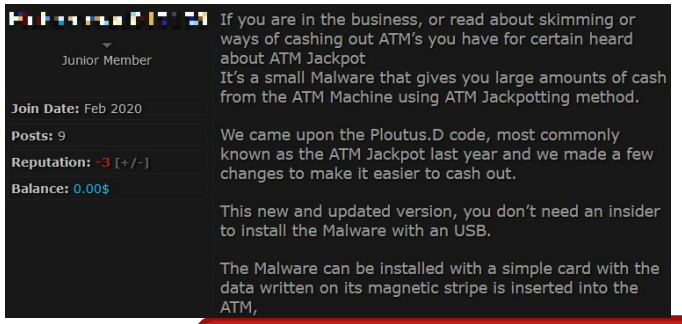
Mercati aperti e mercati chiusi

- A seconda di ciò che si vende esistono due (tre?) diversi tipi di mercati nel dark web, aperti e chiusi
- La differenza principale è nel livello di affidabilità che sia il cliente che il venditore devono avere per poter partecipare al mercato.
- Nei mercati aperti sia il cliente che il venditore non badano alla reputazione. La truffa è dietro l'angolo e non c'è nessuna garanzia sull'affidiabilità di ciò che si compra.
- Tipicamente in questo tipo di mercati si vendono dati come:
  - Carte di credito
  - Device hackerati
  - Materiale sensibile
  - Ecc.

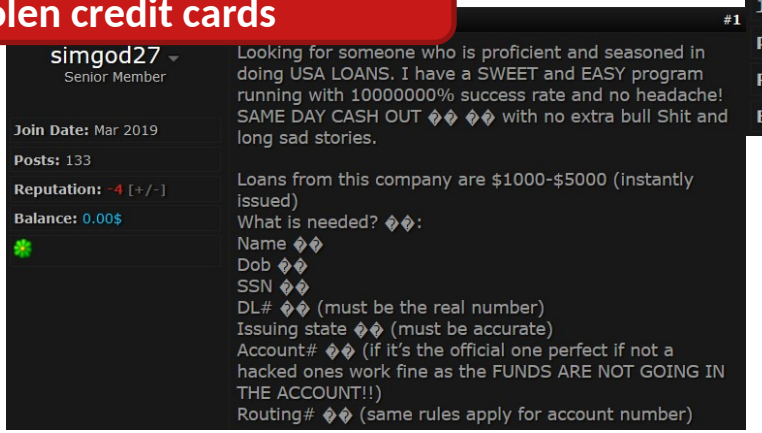
Mercati aperti

- L'affidabilità dei venditori è nulla e il reward è minimo perché molto spesso il materiale è venduto a prezzi estremamenti.

- Il venditore punta sul fatto che non potrà essere denunciato da chi cerca di comprare qualcosa di illegale e dal fatto che può ricrearsi un nuovo profilo molto facilmente essendo i mercati aperti non legati alla fiducia



You can't call the cops for being scammed when you're buying stolen credit cards



Michele Campobasso and Luca Allodi, *Impersonation-as-a-Service: Characterizing the Emerging Criminal Infrastructure for User Impersonation at Scale*, **In the Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS '20)**

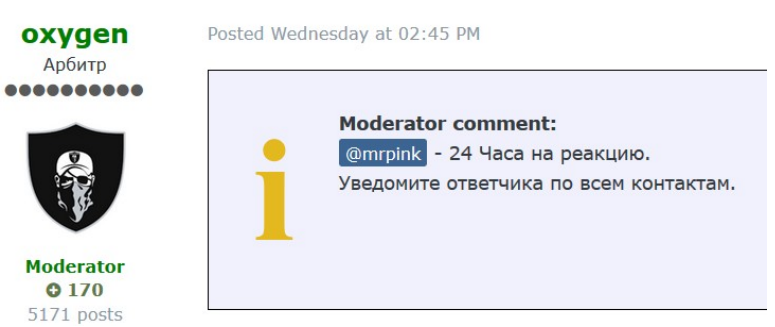
Mercati chiusi

- Nei mercati chiusi il discorso è molto diverso
- Sia venditori che clienti sono profilati a seconda della loro affidabilità
- I venditori in questo caso vendono informazioni “disruptive” come ad esempio:
  - 0-days
  - Leak di attacchi a compagnie importanti
  - Intere profilazioni di centinaia di utenti
  - Chiavi di decifratura ransomware

- In questi mercati i venditori non posso fare truffe. Sono molto di meno ad avere questo tipo di informazioni (rispetto ai mercati aperti) e la loro reputazione crollerebbe molto facilmente
- Per questo motivo esistono veri e proprio meccanismi di feedback e voting, con tanto di chat - customer service!
- Spesso, come ad esempio nel caso di vendita di leak, è possibile anche fare la richiesta di un “campione” di prova dei dati leakati.

Mercati chiusi

- Lato cliente invece c'è bisogno di provare la propria identità è affidabilità più volte.
- Oltre ad una registrazione può essere infatti necessario dover:
  - Ricevere un invito da più membri del mercato a loro volta affidabili
  - Effettuare delle verifiche sul profilo
  - Ottenere dei voti di approvazione da altri membri.
- Sono gli stessi venditori inoltre a poter richiedere informazioni aggiuntive durante la compravendita.



Michele Campobasso and Luca Allodi, *Impersonation-as-a-Service: Characterizing the Emerging Criminal Infrastructure for User Impersonation at Scale*, **In the Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS '20)**